Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 501929
UNCLASSIFIED	

TITLE

A CFD COMPUTER SIMULATION FOR MODELLING LARGE-SCALE BLAST PROPAGATION \(FBINBLAST\)

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

UNCLASSIFIED

DEFENCE RESEARCH ESTABLISHMENT
CENTRE DE RECHERCHES POUR LA DÉFENSE
VALCARTIER, QUÉBEC

DREV - R-9612

Unlimited Distribution / Distribution illimitée

A CFD COMPUTER SIMULATION FOR MODELLING LARGE-SCALE BLAST PROPAGATION (FBINBLAST)

by

C. Fortier and F. Beaumont *

February/février 1997

* École Polytechnique de Montréal

Approved by/approuvé par

Chief Scientist/Scientifique en chef

11 Feb '97

Date

SANS CLASSIFICATION

ABSTRACT

This report describes a large-scale medium-resolution computer simulation of blast propagation inside a structure with many compartments, following the explosion of a conventional warhead inside one of the compartments. The simulation is an extension of the quasi-static blast-propagation simulation INBLAST, which is part of the General Vulnerability Assessment Model. In contrast with the 1-D or 0-D former INBLAST, the new simulation uses a 3-D third-order finite-differences scheme and is able to model the shock wave in addition to the quasi-static pressure. The rupture pressures and the systems' vulnerability are computed in a way that is similar to what was done in the former version. Refinements in the modelling of panel response and rupture have been added in order to better utilize the more precise loading characterization of the new CFD algorithm.

RÉSUMÉ

Ce rapport décrit une simulation par ordinateur, à grande échelle mais de niveau de résolution intermédiaire, destinée à modéliser la propagation de la pression à l'intérieur d'une structure possédant plusieurs compartiments, après l'explosion d'une ogive conventionnelle dans l'un des compartiments. La simulation est une extension de la simulation de propagation quasi statique de souffle INBLAST faisant partie du "General Vulnerability Assessment Model". Contrairement à INBLAST, la nouvelle simulation utilise un schéma numérique 3-D d'ordre 3 et peut modéliser l'onde de choc en plus de la pression quasi statique. La pression de rupture et la vulnérabilité des systèmes sont évaluées de façon similaire à celle d' INBLAST. Certaines améliorations ont été apportées à la modélisation de la réponse et de la rupture des panneaux dans le but d'exploiter au mieux la précision supérieure du nouvel algorithme de dynamique des fluides.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

For many years, DREV's Vulnerability/Lethality Group has been involved in computer modelling of the vulnerability of different land and sea targets and their systems to common threats like blast, fragmentation and fire. These computer simulations are intended to be used mainly for design and procurement purposes (e.g. by DSE for the maritime side), but also occasionally in the operational context. For the study of surface ship vulnerability, the main tool developed and used by DREV is the General Vulnerability Assessment Model (GVAM), which includes a crude quasi-static internal-blast propagation simulation called INBLAST.

Until recently, the DREV vulnerability simulations were aimed at solving large-scale problems: for example, a whole ship was considered. This implied a rather crude modelling and often forced DREV to consider only the most likely threats and targets. But the context, especially typical threats and targets, has changed. The present involvement of the Canadian Forces in new roles, such as UN missions, implies a need for protecting non-armored or lightly armored land vehicles and unprotected personnel against different blast threats like mines. The CF want to assess the vulnerability of unprotected personnel to blast, acceleration, bullets and fragments, in the frame of the IPCE project and in the frame of the multi-phase munitions assessment project. Predicting damage due to local vehicle acceleration and flying objects requires more accurate tools than usual for blast vulnerability simulations. Since the scale of the studies has been somewhat reduced, the need to sacrifice accuracy to execution speed is less obvious than it was.

All these reasons convinced DREV that there was a place for a blast-propagation simulation with a level-of-detail intermediate between crude large-scale simulations like INBLAST and detailed but small-scale simulations like IFSAS, the latter being a precise but extremely time-consuming Finite-Element/Differences CFD and Structure-Response package used by DREV, DSE and DRES for studying local problems.

To address this requirement, DREV contracted with l'École Polytechnique de Montréal to develop a new simulation, FBINBLAST. In contrast to the 1-D or 0-D former INBLAST program, the new simulation uses a 3-D third-order finite-difference scheme to simulate the propagation of the explosion from compartment to compartment. FBINBLAST is also able to model the shock wave in addition to the quasi-static pressure. Component and system vulnerability is computed in a way that is similar to what was done in the former version. Refinements in the modelling of panel response and rupture have been added in order to better utilize the more precise loading characterization of the new CFD algorithm.

This report summarizes the FBINBLAST logic, describes the program strengths and weaknesses and shows to the potential user how to prepare FBINBLAST runs and how to interpret the output. It does not attempt to describe in detail the FBINBLAST algorithms, since this has already been done in Ref. 3.

DREV and DSE intend to use the new simulation instead of the former INBLAST simulation in cases where execution-time constraints justify it.

1.0 INTRODUCTION

DREV's Vulnerability/Lethality Group is involved in computer modelling of the vulnerability of different land and sea targets and their systems to common threats like blast, fragmentation and fire. These computer simulations are intended to be used mainly for design and procurement purposes, but also, occasionally, in an operational context, as illustrated in Ref. 7. For the study of surface ship vulnerability, the main tool developed and used by DREV is the *General Vulnerability Assessment Model, (GVAM* , Ref. 8), that includes a crude quasi-static internal-blast propagation simulation called *INBLAST* (Refs. 10, 11 and 13). Until recently, the simulations were aimed at solving large-scale problems; for example a whole ship was the typical target in a maritime vulnerability context . This implied a rather crude modelling and often forced DREV to limit itself to the most likely threats and targets.

The current involvement of the Canadian Forces in new roles, such as UN missions, implies a need for protecting non-armored or lightly armored land and possibly maritime vehicles, as well as unprotected personnel, against different blast threats like mines. The vulnerability of personnel to blast can now be predicted with much more accuracy than in the past by taking into account the whole pressure history data rather than only the peak pressure and impulse. Predicting damage due to local vehicle acceleration and flying objects requires more accurate tools than was usual for blast vulnerability simulations. Since the scale of the studies has been somewhat reduced, the need to sacrifice accuracy to execution speed is less obvious than it was. The above reasons, added to some occasional modelling problems and inconsistencies (noted by DREV and DSE) lead DREV to dedicate some effort recently to develop more accurate simulations of blast and shock propagation and of structure response, as illustrated in Refs. 2 and 9. In this context, it was felt that there was a place for a blast-propagation simulation featuring a level-of-detail intermediate between crude large-scale simulations like *INBLAST* and other detailed but small-scale engineering-design simulations, like the *IFSAS* CFD simulation used by DREV, DRES and DSE, that tend to have a high degree of precision but are limited to very simple targets unless the user is ready to spend days of execution time on a typical workstation.

To reach this goal, François Beaumont, a student from École Polytechnique de Montréal, has collaborated with DREV during two consecutive summers on improving *INBLAST* (as documented in Ref. 2). He had then produced a new *INBLAST* -based program called *FBINBLAST*. Following this work, F. Beaumont worked on a master's degree project on the matter, under a two-year DREV contract to École Polytechnique de Montréal. The output of this work is the final version of *FBINBLAST* (Refs. 3, 4 and 5) which is the object of this report.

In contrast with the 0 or 1-dimensional analytical approach used by former versions of *INBLAST* and *FBINBLAST*, the new simulation uses a 3-Dimensional third-order finite-volume scheme to simulate the propagation of the explosion from compartment to compartment . This scheme allows precise modelling of blast propagation and its multiple reflections and of the shock wave, in addition to the quasi-static pressure propagation that was already modelled by the former versions. Another difference is that *INBLAST* did not model properly the venting between compartments in the sense that it assumed that after a panel rupture, equilibrium was reached instantly between the two adjacent compartments. The first version of *FBINBLAST* was modelling this venting as a 1-D isentropic flow through a converging-diverging nozzle, which was an improvement. The current version models this by solving the 3-D Euler fluid dynamics equations which should give much better results in principle.

The different panel rupture probabilities, the rupture pressures and the systems' vulnerability are computed as was done in the former version, except that refinements in the modelling of panel response and rupture have been added in order to better utilize the more precise loading characterization of the new CFD algorithm.

The present document summarizes the *FBINBLAST* logic in Chapter 2. Chapter 3 reports on the validation and it evaluates the program strengths and weaknesses. Chapter 4 describes minor modifications done by DREV on the original École Polytechnique version. The appendices constitute a copy of an on-line *FBINBLAST* user manual; for the time-constrained user, reading them is probably enough. The present report does not attempt to describe in detail the *FBINBLAST* algorithms (physics and computer implementation), since this has already been done in the different papers produced by F. Beaumont (Refs. 3, 4 and 5). The emphasis is rather put on the interpretation of the results.

The work on the final version of *FBINBLAST* has essentially been done from September 1993 to September 1995 in the framework of a master's degree project at École Polytechnique de Montréal, sponsored by DREV under PSC 2311B-13A (Vulnerability

Studies). The student worked under the supervision of Professor J. Y. Trépanier. F. Beaumont also worked sporadically from September 1995 to March 1996, after obtaining his degree, on the calibration of the simulation and on the report appearing in Ref. 3, under the same PSC.

2.0 LOGIC OF FBINBLAST

2.1 Program Flow

The program takes as input a block description file (Fig. 1), a panel-description file and an attack-description file described in detail in the appendices. These files are the same as for the *INBLST GVAM* module, except for the attack file that contains a special record. The panels are extracted from a block-description of the ship, which contains information about their dimensions, thickness, material, rupture-pressure and principal eigen-frequency. The attack file mainly contains the attack scenario. This consists of the compartment where the (warhead) explosion occurs, the relative location of the charge in the compartment and its equivalent weight of TNT ; the special record contains the simulation end-time plus the threshold probability of panel rupture over which a panel is actually to be considered ruptured. As will be seen, this last information has an important effect on the modelling of blast propagation.

Starting from this input, *FBINBLAST* initiates an explosion in the compartment of origin, as a high mass and energy-density sphere of air. It then models its propagation by solving the Euler equations, as explained in Ref. 4. Shock, reflections and quasi-static pressure are precisely resolved numerically using a 3rd order Essentially Non-Oscillatory (ENO3) finite volume scheme (see e.g. Ref. 15). This scheme is a compromise between more precise Total Variation Diminishing (TVD) schemes and cruder algorithms; it essentially maintains the precision of the TVD scheme without a too high cost in execution speed. External venting is also modelled. Adapted gridding is used. Panel rupture is modelled in a way similar to what was done in former *INBLAST*, except that an algorithm (Ref. 5) based on the position of each panel cell with respect to the panel borders is employed. This allows one to take advantage of the better precision of the fluid dynamics calculations (the former algorithm was 1-D or order 1); this algorithm is explained in more details below.

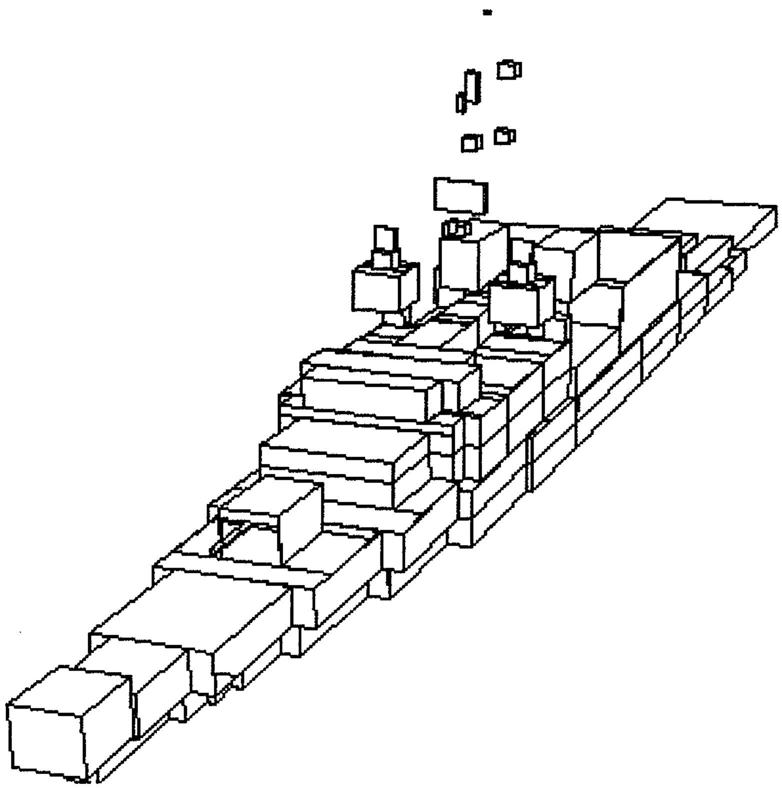

FIGURE 1 - Contents of a block-file (crude model of a DDH-280 destroyer)

The output, as concerns primary (as opposed to component) damage, is done in two types of files having the same format as the former *INBLAST* output files (see Appendix A). One of the files is intended to describe panel events; output is done at every panel rupture and consists of the corrected (Ref. 5) pressure acting on the panel, the current time, the current and cumulative failure probabilities and the IDs of the two adjacent compartments affected by the rupture (only one if the panel is external). The second file is to be seen as containing compartment-related information and is appropriate for deck-by-deck display by the standard *GVAM* graphical program *INGRAPH* or for subsequent component-damage calculations (Appendix A). In the latter case, each invaded compartment appears only once and the output pressure is the maximum over time of the mean (over cells) compartment overpressure. In both cases, the cumulative probability is to be interpreted as the probability of invasion of the destination compartment (the other compartment bordered by the panel being the source compartment).

The exact input and output file format and interpretation is detailed in the appendices, as well as the use of the *INGRAPH* display program, the *DAMAGE* component-damage program and the *SURV* system-damage simulation.

The primary (compartment level) damage output by *FBINBLAST* is then normally (although it is not mandatory) used, in conjunction with a description of each of the components constituting the target's systems, to calculate the component damage; this is done by the *DAMAGE* program. The next step is normally to calculate the damage to different systems from the *DAMAGE* output and from a description of the systems as networks of components.

Note that the components and systems can be described by using a graphically oriented program (*FTA*, Figs. 4 and 5); this program will output a system-description file to be used at run-time. *DAMAGE, FTA* and *SURV* are general-purpose *GVAM* modules distinct from *FBINBLAST* , but *FBINBLAST* can also calculate component and system vulnerability by itself, as was the case in the pre-*GVAM-III* versions of *GVAM*. "System" in this context are to be interpreted as Primary Mission Area (PMAs, like Anti-Air Warfare, Mobility, etc.) that are themselves series of networked components (sub-PMAs). The appendices explain how to use these programs (see also Ref. 8 for more details).

As is usual for GVAM programs, the damage from the following threats (*GVAM* modules) may be combined:

- *Internal blast (FBINBLST /INBLSTprogram)*
- *External blast (EXBLST program)*
- *Fragmentation (FRAGMT /FRAGMT2 program)*
- *Fire propagation (FIRE program).*

Combination in this context means that the component damage from each threat can be combined independently by *DAMAGE*, and by ricochet, by *SURV*, as explained in Appendix A.

2.2 Treatment of Panel Ruptures

2.2.1 Physical Model

The former *INBLST* simulation was modelling rupture very roughly as the dynamic loading of a harmonic spring (the panel) with a uniform pressure applied to it, as done in Ref. 14. The pressure was subsequently uniformly distributed to all the ruptured compartments, thus considering an ever-expanding compartment of origin. To compensate for this coarseness, a second time-related condition was imposed to rupture the compartment. A certain rupture time had to be reached for a failure to be decided upon. It corresponded to the loading time of the source compartment, calculated by considering the loading process as an isentropic flow through a converging-diverging channel, plus a delay accounting for the panel response time, plus a final delay for modelling the crack propagation time.

Since *FBINBLAST* maintains information on several cells adjacent to a loaded panel, it is possible in theory to model the rupture process in a more precise way. Similarly, since *FBINBLAST* models shocks and reflections in addition to the quasi-static pressure, the relevance of the above procedure appeared doubtful; in particular, the compartment loading delay did not make sense and is not considered anymore in the new version, although the other two delays still are. Refefence 5 describes in detail the work that has been done to adapt the modelling of rupture to the new context.

The new rupture model, essentially lumps the contributions of the different cells adjacent to the current panel into a uniform dynamic loading on the panel. Before being summed, the forces acting on each cell are weighted following their distances to the panel edges in order to account for their relative influence on the maximal bending moment. The panels are assumed clamped, so that the maximal moments will occur on the edges. As before, the panels will not be allowed to rupture before a certain delay has elapsed, except that this time *FBINBLAST* is assumed to take into account the compartment loading delay by itself. A drawback of this approach is that it assumes that panels have a height-to-length ratio around 1, which is not always the case. The algorithm also assumes a sufficient number of cells per panels (say 100).

When a panel breaks, *FBINBLAST* uses mirror cells (3 rows) at the corresponding compartment border. When the grid is not uniform, this may create some special numerical stability problems. A compartment wall may comprise several sub-panels; the rupture of such a sub-panel may create new edges thought to be the cause of the problem. The situation is further complicated by the fact that the grid resolution must be taken quite coarse in order to maintain a reasonable execution time. Without going into details, which are explained in Ref. 4, this has been solved by the adjunction of an artificial viscosity term. It has been checked that this did not affect the quality of the output in any noticeable way.

2.2.2 Logical Rupture

The above rupture model is (implicitly) based on a maximal bending criteria. It is not 100% clear that this is the most relevant criteria in all circumstances; a maximal shear or energy criteria could also have been used. The types of boundary conditions may also vary and the 'clamped' assumption may not always be the most realistic one. Hard-to-control factors like soldering and manufacturing imperfections and corrosion can also affect the rupture. All these factors lead to consider rupture as a more or less stochastic event. It must also be realized that errors (approximations), sometimes noticeable, are made when modelling the applied pressure as well as when calculating the panel dynamic rupture pressure. Since the rupture or not of a panel may have a tremendous influence on the future propagation of the blast, some way must be found to give a less deterministic meaning to ' *panel rupture*' . In *FBINBLAST*, as is the case in the former *INBLST* simulation, this is done as explained below.

To decide about panel rupture, *FBINBLAST* passes the ratio, R, of the observed corrected pressure to the theoretical rupture pressure (as calculated by program *gvmfm1*) to a log-normal probability distribution, *gfunc2*, that outputs a rupture probability corresponding to the current value of R. To decide if the blast invades the compartment or not, which indirectly determines if possible ruptures of new panels in this compartment will be studied or not, the simulation compares the output value of *gfunc2* with a certain threshold (p specified in the attack file) and proceeds to the rupture if the value is higher, provided that the response delay has elapsed. When $R = 1$, the actual pressure has reached the theoretical rupture pressure; *gfunc2* then outputs *0.5* and increases or decreases very fast right or left, respectively, of this value. Such a log-normal algorithm is also used when estimating the component damage (program *DAMAGE*, routine *gfunc*). The graph of *gfunc*/*gfunc2* corresponds to a S-shaped increasing curve having value *0* at left of *0* and tending to *1* at *infinity*. The *COV* parameter of these routines controls the abruptness of the slope. *COV* may be interpreted as the square root of $ERRP^2+EERR^2$

where *ERRP* and *ERRR* are the estimated relative errors (standard deviations or half *68%* confidence interval) on the rupture pressure and actual pressure, respectively. The two relative errors are assumed to be independent, normally distributed, with mean *0*. The *gfunc2* function is also used to randomize the crack-propagation time. See Ref. 8, "Chapter Damage" for more information about *gfunc2*.

The above-mentioned threshold probability *p* is to be seen as a safety factor for the above algorithm. The value *0.5* would be used if the rupture pressure and actual pressure calculations were considered perfect; then the rupture would occur at the precise moment where the theoretical rupture conditions have been reached. For vulnerability studies, a smaller number should be used in case of doubt; rupture would then occur as soon as conditions make the probability of rupture greater than *p*. For lethality studies, a greater number should probably be used.

Obviously, the (cumulative or not) probabilities of compartment invasion, that are output in the *FBINBLAST* output files, will always be higher than the value of *p*. Experience shows that they are usually much higher, due to the abruptness of the log-normal damage function used.

2.2.3 Cumulative Probability of Invasion

Given the fact that *INBLST* was simply extending the compartment of origin, it was relatively easy to define the cumulative probability of invasion of a compartment; it corresponded to the cumulative rupture probability of a single panel, the first and only one to break in the compartment. With *FBINBLAST*, several panels of the same compartment may break, and at different moments. *PE(i,t)* = cumulative probability of compartment *i* having been invaded by blast at time *t* is then defined recursively as follows:

PE(i,t) = 0 until the rupture probability threshold has been reached
for at least one of the compartment's panel

At each new logical rupture, say across panel *p*, limiting compartments *j* and *i*, then *PE(i,t)* becomes:

$$PE(i,t-dt) + PE(j,t-dt) \times Q(p,t)\,(1 - PE(i,t-dt))$$

where *Q(p,t)* = current rupture probability of *p*, and where *dt* is the simulation step. This amounts to consider that the current probability of compartment invasion at time *t* is the probability of the union of the two following events:

A = *{i has already been invaded}*
B = *{i is presently being invaded across a new panel}*.

Note that the probability of B is nothing but the probability that the neighbor j has already been invaded multiplied by the probability that the panel between j and i breaks, as indicated by the formula. A careful reader would have noticed that the calculation assumed that A and B were independent, which is false strictly speaking, but one would wonder how to treat the problem without this assumption that is incidentally typical in vulnerability studies.

2.3 External Venting

External venting is modelled by considering that the exterior of the target is a large air tank at the (constant) atmospheric pressure and temperature. The flow direction at the ruptured compartment's external border is maintained constant and is assigned the specific mass and momentum corresponding to the border row of cells (order 0 approximation); the specific energy is calculated from Bernouilli's equation.

2.4 Computer Platform

The current version of *FBINBLAST* runs on a Sparc Station II (Sun RISC) UNIX workstation. It is a standard FORTRAN77 program, that inputs and outputs in ASCII files, usually extensively commented. The same is true for the *DAMAGE* and *SURV GVAM* programs.

The *fta* system-modelling program, and the *INGRAPH* deck-by-deck display program require the XWindows/XView interface (dynamically linked). If they are to be recompiled, the XGL graphic library (not delivered with the system) is also required. In addition, the (public-domain) xps and xvps PostScript interface libraries are necessary for recompiling *INGRAPH*.

3.0 VALIDATION AND CALIBRATION

3.1 Single Compartment

The ENO3 scheme has been implemented in three steps corresponding to 1, 2 and 3 dimensions, respectively. At each step, the scheme has been compared with known academic or experimental CFD problems. A classical shock-tube problem has been the 1-D test case. A corner reflection problem whose output appears in Fig. 2 has been used for the 2-D algorithm. The precision of the scheme has been compared with success to interferograms and to the output of other schemes (e.g. TVD). Figure 3 illustrates the type of resolution afforded by the ENO3 scheme for the case of an explosion propagated through a hole, at three equally spaced time steps.

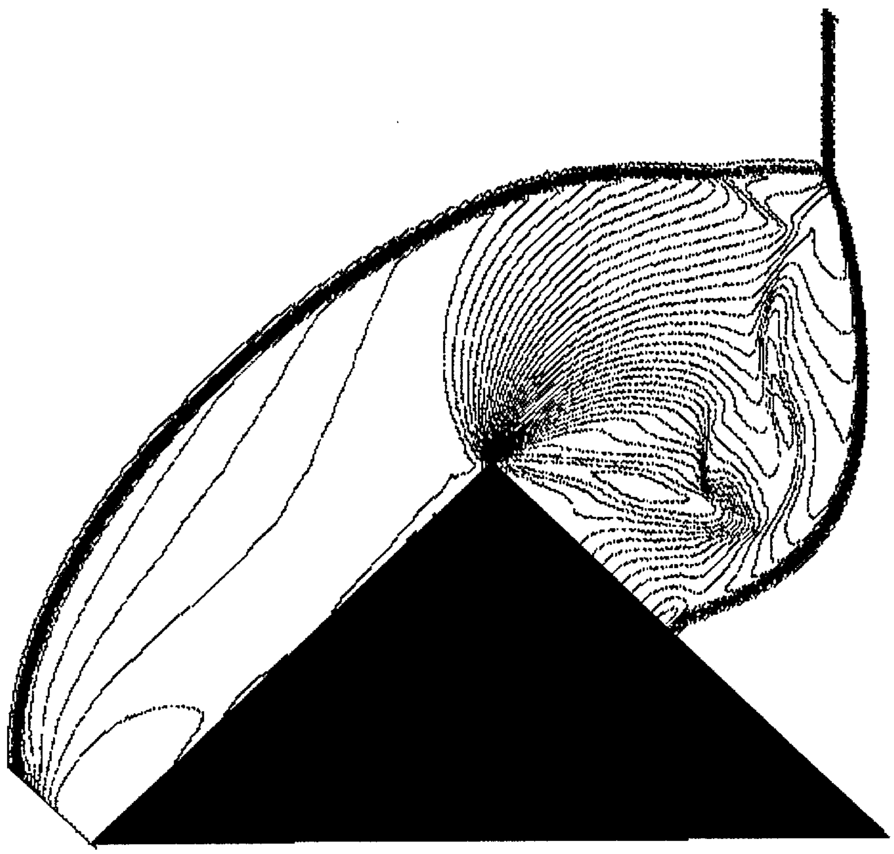

FIGURE 2 - Iso-density lines for corner problem (solved using ENO3)

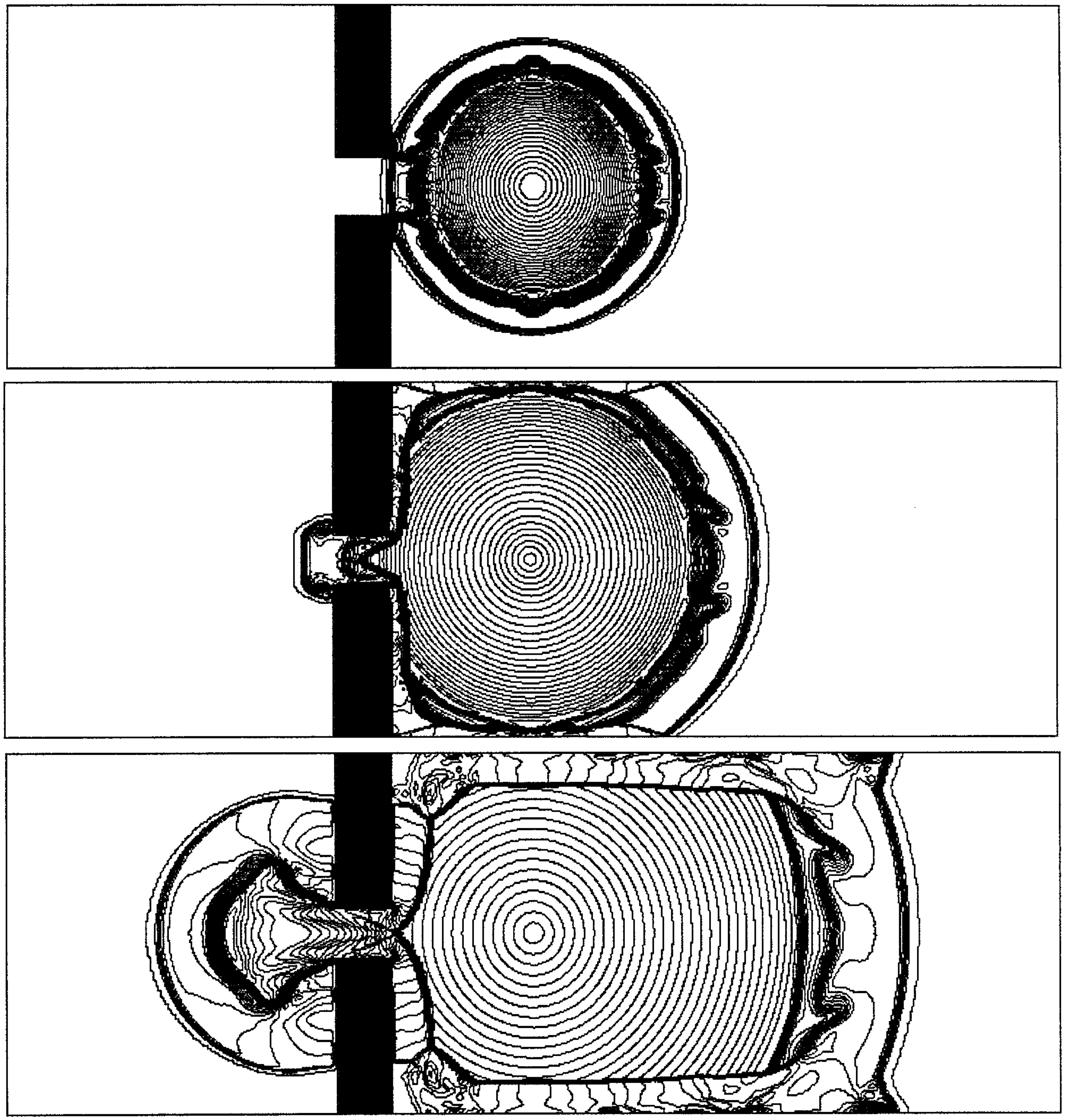

FIGURE 3 - Iso-Mach lines for propagation of an explosion across a hole

In 3-D, the test problem described in Ref. 6 has been used; it consists in exploding 1 lb of C4 in the center of a rectangular bunker. The pressure traces output by the ENO3 scheme proved to correlate very well with the actual gauges output, even in the corners, which proves that the scheme treats the reflections correctly. The tests also in the case of certain grids; for this reason, *FBINBLAST* allows the use of an ENO2 scheme if desired (after recompiling). Despite appearances, the latter problem is a multi-block one, except that the holes between the blocks are fixed (do not involve any panel-rupture process). In this sense, one is led to conclude from this trial that the multi-block implementation works fine. All these trials and more are documented in Ref. 4.

3.2 Multi-Block Algorithms

The quality of the multi-block implementation is obviously strongly dependent on the modelling of rupture, since this decides if the blast invades a compartment or not, which in turn influences the propagation to the next layer of compartments, etc. After some comparisons with multi-compartment blast propagation trials, Ref. 4 concludes that the new simulation largely overestimates the damage. The explanation proposed for this fact is that the modelling of rupture is faulty, presumably because it neglects to account for the energy spent by the expanding gas for breaking the panels. The rupture time calculation also appears faulty. As noted in the next Chapter, DREV believes that the faulty damage-algorithm *gfunc2* used to filter the actual-pressure to rupture-pressure ratio may be responsible, at least partly, for the faulty results. It is thought that a new comparison with actual tests, using the corrected *gfunc2*, could possibly output much better results, although the rupture-time calculation problem would still remain.

DREV also checked carefully, using numerous scenarios that the simulation logic was not faulty even in limit conditions. These trials made apparent a few problems, not related to the numerical scheme, that are explained in the next chapter. After their solution the program performed as expected.

3.3 Interfacing with Other Modules

In principle the component and system damage calculations done by programs *DAMAGE* and *SURV* did not have to be checked since these are the normal *GVAM* modules that post-process the *FBINBLAST* instead of the *INBLAST* output . It has nevertheless been checked that these programs worked properly in the sense that they interfaced correctly with *FBINBLAST*.

The graphical display of the results contained in the *FBINBLAST* display file (*Dispfile.INR* that contains the compartment-related information) by module *INGRAPH* has also been found to be correct.

3.4 Execution Time

Execution time is about 30 min for the typical 0.06 s attack of 400N of TNT in compartment 20 of block model *cube.TBM* (27 compartments on 3 identical decks). This indicates that a similar simulation on a reasonably complex model, like block model *DDH280.TBM*, would last hours at best. It appears hardly thinkable to improve this figure by reducing the resolution; precision would cease being acceptable. In this regard, the performance of *FBINBLAST* is a little disappointing, but may still be considered acceptable given the large scale of the problems studied compared to other CFD simulations.

3.5 Remarks

There is a problem with the panel files that the contractor used for the trials. The problem occurs when modifying the files, even very slightly. The source of the bug may be the change of platform (IBM Risc -> Sun Sparc); it may possibly be attributed to different conventions concerning carriage returns and line feeds for marking the end of lines. This problem does not occur when the files are generated using *gvmfm1* (see Appendix A) on a Sparc machine.

The simulation end-time, especially if it is very short, may have a tremendous influence on the maximum (over time) of the mean overpressure in a compartment as output in *Dispfile.INR*. The reason is that the simulation could end much before the pressure had any chance to fill a newly invaded compartment; the output pressure could be lower than the maximum theoretically reachable. The user should keep this in mind when using *FBINBLAST*.

4.0 MODIFICATIONS TO THE ORIGINAL PROGRAM

The initial version delivered by the contractor in February 1996 had a number of inconsistencies concerning the general program logic (as opposed to the finite-volume algorithms) that have been corrected.

4.1 Interpretation Program Arguments

Routine *inblst.for* was not reading correctly the program arguments (I/O file names), which prevented calling *FBINBLAST* with the syntax:

inbl [file1.INT file2.TBM file3.ATI file4.INR]

This is now solved.

4.2 Repetition of Destination Compartments in Dispfile.INR

Routine *graphadj.for* was faulty in the sense that it output records corresponding to the same destination compartment several times. It now performs as indicated in Appendix A.

4.3 Error in the Calculation of Cumulative Probabilities of Invasion

Routine *addition.for* did not calculate properly the cumulative probability of compartment invasion; the output was roughly corresponding to the probability of invasion across the last panel of the compartment that possibly failed. This has been corrected in order that the probability is now calculated as explained in Appendix A.

4.4 Faulty Damage Algorithm

In *FBINBLAST*, the original *INBLAST gfunc* routine (log-normal damage algorithm) had been replaced with a strange function named *gfunc2*. It featured a logistic damage algorithm such that the output probability of panel break was *0.7* when the rupture pressure was reached; it should be *0.5*. To correct this anomaly, the *gfunc2* routine has subsequently been replaced by DREV with a simplified version of the former *INBLAST gfunc* routine. It is believed that the faulty *gfunc2* was one (if not the main) cause explaining the systematic damage overestimation mentioned in paragraph 2 of the conclusion of Ref. 3.

4.5 Modifications Made to Routines *READM* and *SIMUL*

Minor modifications have been done to routines *READM* (reads the panel file) and *SIMUL* (program main loop).

5.0 DISCUSSION ON SPECIAL PROBLEMS

Problems have been noted for the multi-block implementation when rupture is taken into account. Despite the work done for improving the modelling of rupture, the new simulation seems to overestimate the damage. Reference 3 attributes the discrepancies with live trials to a faulty modelling of panel rupture. Although an error in the log-normal damage algorithm that caused an overestimation of the rupture probability has been corrected, the rupture calculations are probably still more or less reliable due to the dubious response time calculations. One solution to this, as noted in Ref. 5, would be coupling the loading (CFD) simulation with a finite-element panel-response algorithm for predicting panel rupture as well as hole-area (the current simulation assumes that the whole panel collapses). Unfortunately, this could considerably slowdown the (already very slow) *FBINBLAST* simulation.

Another cause of the poor rupture modelling mentioned above is the implicit use of a maximal bending criteria. It is not clear that this is the most relevant criteria in all circumstances; maximal shear or energy criteria could also have been used. The types of boundary conditions may also vary and the 'clamped' assumption may not always be the most realistic one. Another panel damage mechanism not taken into account, which appeared clearly in recent trials, is the kinetic energy of already collapsed panels. It is also unclear if the crack-propagation delay used in modelling the rupture time still makes sense in this context. Finally, panel rupture predictions can be considered as inherently unreliable (stochastic) due to hard-to-control factors like soldering imperfections, corrosion, fatigue and faulty manufacturing processes that could possibly be taken into account, but not completely.

A new complication also arises in *FBINBLAST*. If one uses a smaller than *0.5* probability threshold for taking a decision about panel rupture, presumably with the idea of being conservative in a vulnerability study, the estimate may lead to damage overestimation, at least in some cases. Rupturing a panel that would not do so in reality may allow the simulated blast to use a path starting with this panel where damage could be higher (or even lower) than if rupture occurs. Without rupture for example, the pressure could have been confined to the source compartment or another 'harmless' path could have been initiated across a second panel. A similar remark is valid if one overestimates the resistance of the first panel (i.e. for lethality studies); the pressure accumulation in the compartment may cause the rupture of an unexpected wall, opening the way to a higher damage path. This complication also occurred in the former simulation, but to a lesser extent because invasion occurred only once per compartment ; this is because *INBLAST* was simply expanding the compartment of origin. In this aspect, *INBLAST* had certain characteristics of a Damage Radius approach; such a simulation, although cruder, features less interpretation problems than one trying to follow possibly self-crossing blast paths like *FBINBLAST*.

A final difficulty appears for calculating the cumulative damage probability of compartment invasion. One has to assume the independence of two events: new panel rupture and already existing invasion. This assumption is not entirely justified, but most vulnerability simulations have to make similar assumptions. This question did not arise with the former *INBLAST* due to its logic.

6.0 CONCLUSION

There is no doubt that solving the 3-D Euler equations in *FBINBLAST* instead of using a 1 or 0-Dimensional model as in the former *INBLAST* is a significant improvement in realism. The scheme used has been proven reliable for modelling the blast propagation as long as panel rupture is not taken into account , as proven by numerous test cases, some single-block, some multi-block. The scheme is reasonably fast although the new simulation is slower than expected (hours versus seconds for the former *INBLAST*). The ability to model the shock and its reflections is also an important improvement over the former version that featured only quasi-static pressure propagation.

The modelling of the rupture process is nevertheless still too crude (as was the case in *INBLST*) and may lead to false conclusions concerning the extent of the damage. Some more work concerning this aspect would be useful .

7.0 REFERENCES

1. "Structures to resist the effects of accidental explosions", Department of Defense Explosives Safety Board Alexandria, Virginia, 1990, UNCLASSIFIED (no author).

2. Beaumont, F., "Description du modèle simulant l'évolution de la pression interne dans un navire, suite à un explosion: FBINBLAST", DREV M-3181/94, février 1994, SANS CLASSIFICATION

3. Beaumont, F., "Description du modèle simulant l'évolution de la pression interne dans un navire, suite à un explosion: FBINBLAST" (Rapport final), École Polytechnique de Montréal, février 1996.

4. Beaumont, F., "Modélisation des multiples réflexions d'une onde de choc générée par une explosion dans un compartiment clos", École Polytechnique de Montréal, juin 1995.

5. Bédard, S., "Détermination des conditions de rupture d'une plaque rectangulaire encastrée soumise à un chargement", École Polytechnique de Montréal, avril 1995.

6. Chan, P. C. and H. H. Klein. "A Study of Blast Effects Inside an Enclosure.", Transactions of the ASME Vol. 116 (September 1994): pp. 450-455.

7. Czaban, Z.J., Belcher, M., "Report on GVAM Simulation of Exocet Attacks Against DDH-280", DNASE 5-5, NDHQ, Ottawa, February 1991, SECRET

8. Fortier, C., "General Vulnerability Assessment Model (GVAM-III Version) User Manual," DREV TM-9436, February 1995, UNCLASSIFIED

9. Fortier, C., "New Panel-Rupture algorithms and Other Modifications to Exblast", DREV, internal document, September 1992, UNCLASSIFIED

10. Gass, N. and Scarlett, P. F., "Vulnerability Assessment of a Generic Frigate Due to Airblast", DREV M-2601/82, August 1982, UNCLASSIFIED

11. Gass, N. , "Internal Blast Damage Model". TTCP W-1 KTA-10 meeting, Portsmouth, England, 1981. UNCLASSIFIED

12. Griffel, W., "Handbook of Formulas for Stress and Strain", Frederick Ungar Publishing, New-York, 1966.

13. Proctor, J. F. "Internal Blast Mechanisms Computer Program", Naval Ordnance Laboratory, NOLTR 72-231, August 1972, UNCLASSIFIED

14. "TM5-855 (Fundamentals of Protective Design for Conventional Weapons)", Department of the Army, US Army Corps of Engineers, Washington, 1989, UNCLASSIFIED

15. Yang, J. Y. and C. A. Hsu . "High-Resolution, Nonoscillatory Schemes for Unsteady Compressible Flows." AIAA Journal Vol. 30 (No. 6, June 1992).

APPENDIX A

ON-LINE FBINBLAST USER MANUAL

1. SUMMARY RUNNNING INSTRUCTIONS

1.1 Main Program

go to the directory containing the I/O files and type:

inbl [file1.INT file2.TBM file3.ATI file4.INR]

The complete path of *inbl* (= FBInblast) should be included when the current directory is not the one containing the application.

file1.INT is the panel description file,
file2.TBM is the block-description file from which the above has been generated
file3.ATI is the attack file
file4.INR is the output file

The formats of the ".TBM", ".INT" , ".INR" and "ATI" files are explained in the next section and examples are available in directory *./Templates* ; see also Ref. 8 for more details. Normally, the user should generate the ".INT" file from a ".TBN" file, itself generated from a ".TBM", both using the *gvmfm1* utility documented in Appendix B and in Ref. 8 for more details.

If no argument is given to the *inbl* command, *FBINBLAST* expects to find the file names in records identified by a 2-letter code of a file named *gvam.CUR*; the latter must be in the calling directory. The conventions are illustrated below:

EA file3 *! INBLST attack file (.ATI):*
EB file1 *! "ship" internal panels (.INT):*
EE file2 *! "ship" three dimens. block model (.TBM):*
EL file4 *! INBLST vulnerability file (.VLI):*

The file characters following '!' are optional comments. The "EL" record is optional, as opposed to the others. The ".VLI" file contains pre-*GVAM-III* system descriptions as documented in the *INBLST* Chapter of Ref. 8; it is better to forget the corresponding record and to run a component/system vulnerability study, as done in *GVAM-III* ; this is partially documented below and, with more details, in Ref. 8. When the program is called without arguments, the output file is automatically named *attack.INR*, where *attack.ATI* is the name of the attack file.

1.2 Deck by Deck Display

ingraph -r Dispfile.INR &

Dispfile.INR is the default name of the display file. If the user wants to display the components in addition to the compartments in the deck-by-deck-display then it is necessary to change the character chain containing the name of the ".TBM" file, that appears in the ".INT" file header, with the one of the joint component-compartment, ".TBM', file; such a ".TBM" can be generated using the syntax:

showcompon file2.TBM file4.TBM file5.fta

where *file2* and *file4* are the source and output ".TBM" files, and *file5* is the corresponding component and system description file.

1.3 Component-Level Damage

damage PA file6.COMPDAM file5.fta 1 PC Dispfile.INR

file6 is a component damage output file

file5.fta, is a component and systems description file, normally generated interactively using program *fta*, as documented below and in Ref. 8.

1.4 System-Level Damage

surv -r file7.fta-res -f file5.fta -c file6.COMPDAM -inblst

file6 is the output system-damage output file; its records are described below and in Ref. 8.

1.5 Hint

The easiest way to run the programs is to edit the template command files that can be found in directory *./Templates.*

2. INPUT

2.1 Attack

Describe the attack in a ".ATI" file. The latter, that has to be prepared in an editor (possibly by editing a template file like the one in *./Templates*). These files differ from the usual *INBLST* attack files in that they contain an "AL" record; the other records are similar. See the commented example file below for an explanation of the different records.

```
C      The following contains typical example of an "AH" record
C      as it appears in an input data file.
C
C Rec.  Compartment  Attack     Aft    Starboard   Bottom      Area of
C Type  No. of       Direction  to       to          to        Entry Hole
C       Detonation              Fore     Port        Top       (m**2)
C
C AH   13           POR     .7        .4       .2           .1
C
C Notes:
C   The warhead detonation location within a compartment is expressed
C   as a distance ratio between 0 and 1.  E. g. Given
C   that the distance between Aft and Fore panels is 6 meters
C   say, a value of .7 would be 70% of the distance from the Aft panel
C   to the Fore panel.
C
C      The Attack Direction is the one from which the missile is coming
C      not the one it is going to
C
C AL     .1   .5   ! endtime (secs) and minimal probability for considering
C               that a panel is ruptured
C-----------------------------------------------------------------------
AH     20      STA     0.5    0.3    0.2     .2
AI     400                !Warhead weight (Newtons)
AL     .06     .5
```

The second field of the "AL" record is to be seen as a safety factor. The value *0.5* would be used if the rupture pressure and actual pressure calculations were to be considered perfect; then the rupture would occur at the precise moment where the theoretical rupture pressure has been reached. For vulnerability studies, use a smaller number, say *p*, in case of doubt; rupture will then occur as soon as conditions render the probability of rupture greater than *p*. For lethality studies, use a higher number. Note that the ratio of actual pressure to rupture pressure is passed to a log-normal distribution with median at *0.5*; what is meant by 'probability of rupture' is precisely the output of the log-normal.

2.2 Block Geometry

2.2.1 ".TBM" files

A ".TBM" file describing the ship geometry as a series of 3-D rectangular blocks, has to be prepared. This can be done manually in an editor or by using the interactive (graphical) shipeditor program. The exact file format is explained in Appendix B. Fig. 1 illustrates a block model.

2.2.2 ".TBN" files

This file contains essentially the same information as the ".TBM" file, except that the stiffener spacing varies with the panel (it is the same for the six panels defining a compartment in the ".TBM" file). The panel-description files used by *FBINBLAST* (".INT"s) are generated from a ".TBN" rather than a ".TBM" file. Stiffener information is not used by *FBINBLAST*.

The ".TBN" block file is generated semi-automatically from a ".TBM" block file by using the *gvmfm1* program, as explained in Appendix B. The ".TBM" file is later used by *FBINBLAST* but the ".TBN" is not needed except for generating a ".INT" panel file from it. The format is not important for the *FBINBLAST* user but is documented in Ref. 8, Chapter "GVMFM1 FILE CONVERSION PROGRAM", if needed.

2.3 Panels

The panel properties are described in a ".INT" *INBLST* panel file. The panels have normally been extracted from a ".TBN" *INBLST* block file by program *gvmfm1*. Consult Appendix C to know how to run *gvmfm1*. The ".INT" files have the same format as the *INBLST* ".INT" files. See Directory *./Templates* for an example and Appendix D for an explanation of the file contents.

2.4 Components and Systems

A vulnerability file (".VLI") may be used optionally for describing the components or systems if the user wants component and system vulnerability results in addition to the primary damage output of *FBINBLAST*. The file has the format of the old (pre *GVAM-III*) *INBLST* vulnerability file. The format is not explained here since its use is quite outmoded (see Ref. 8 or the example in *./Templates*, if needed).

This is nevertheless not the preferred way to describe the components and systems since the interactive (graphical) program *fta* may be used to prepare the corresponding input, as documented in Ref. 8. This will generate an ".fta" file that is compatible with *FBINBLAST*. Program *fta* may be called with the simple command *fta &* ; the '&' (background) is not mandatory but preferred. The format of the ".fta" files is explained in Ref. 8, if needed, but since the files are normally generated and modified interactively, the user does not need to manipulate them directly in principle. The use of the *fta* program is also explained in Ref. 8; it is indeed quite transparent if components only (as opposed to systems) are defined by the user. Figure 4 illustrates the *Component* dialog of *fta* .

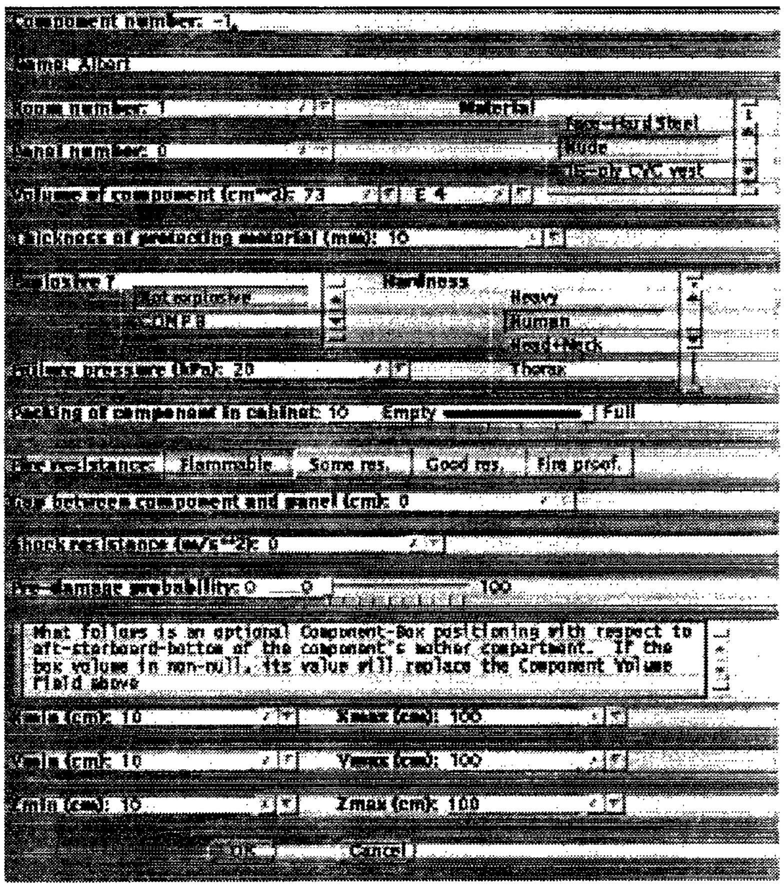

FIGURE 4 - FTA component-definition dialog

The only field related to internal blast in the component-description dialog of program *fta* is labeled *Failure pressure kPa.*. It is the maximal tolerable pressure that the component may suffer. The value should be between *0* and *65000* and defaults to *20*.

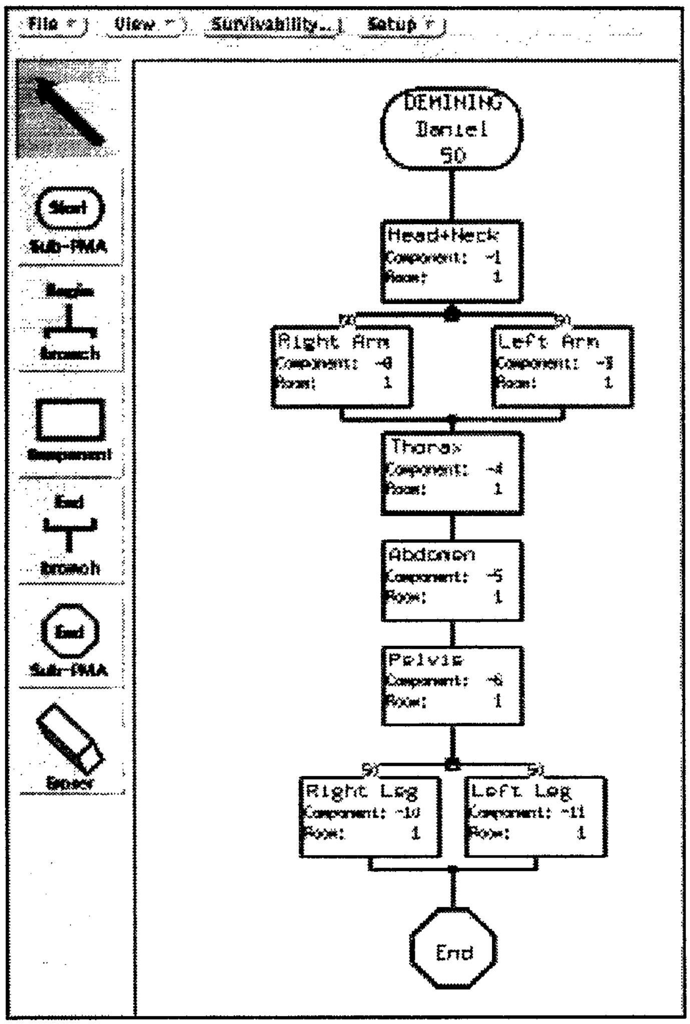

FIGURE 5 - FTA panel with a small system displayed

Systems (PMAs) are networks of components, themselves grouped into sub-systems (sub PMAs), as defined interactively when using program *fta*. The way to define the systems is not explained here (see Ref. 8 and Fig. 5).

The internal-blast-induced damage to components is controlled by the maximal value of the overpressure in the compartment. After the ratio of actual overpressure to rupture pressure has been passed to a log-normal function, the result is multiplied by the cumulative probability of compartment invasion; i.e.

P[damage] = P[damage given invasion] x P[invasion]

Because of this, if a component-damage analysis is required in addition to the *FBINNLAST* primary damage analysis, it is probably better to take the second field of the ".ATI" file's "AL" record as *0.5* ; this amounts to consider that a given panel will break if and only if the conditions of rupture have been met, thus rendering the simulation (quite) deterministic.

2.5 Program Flow

If *FBINBLAST* is to be called without arguments, input the names of the ".INR", ".TBM", ".ATI" and ".VLI" (if applicable) files in file *gvam.CUR*. The latter must be in the directory from which *FBINBLAST* is called and cannot have any other name (see the example in the beginning of this appendix). An example *gvam.CUR* file is also available in directory *./Templates.*

If *FBINBLAST* is to be called with arguments, the easiest way to do it is to edit template command files. Example command files for calling *FBINBLAST*, the component-damage program *DAMAGE* and the component-systems survivability program *SURV* are available in directory *./Templates.* Several program calls may be chained in a single-command file. The syntax for passing arguments is explained in the beginning of the current appendix.

See the GVAM User Manual (Ref. 8) for more explanations on the different file records and their formats.

3. OUTPUT

3.1 Output Files

Detailed output is done in the ".INR" file specified at the program call; this file is referred to as the 'normal' one below. A second ".INR" file, used for display by program *INGRAPH* and for component-damage calculations, is also generated; it is named *Dispfile.INR* by default. When the program is called without arguments, the output file is automatically named *attack.INR*', where *attack.ATT* is the name of the attack file.

The normal ".INR" file is written throughout the simulation, one record being output for each panel such that a pressure high enough to insure panel rupture has been reached, if a sufficient time has elapsed. This file is oriented toward panel, as opposed to compartment, information. All panels that have been pressurized during the simulation will appear once and only once in this file, except possibly for the entry panel that may appear twice. The entry panel will always appear first in the file; this accounts for the missile entry hole.

Dispfile.INR is written in one shot at the end of the simulation. Each record corresponds to a compartment rather than a panel. There is one and only one record per invaded compartment; it corresponds to the blasted compartment panel for which the cumulative probability of compartment invasion is maximal. This also corresponds to the last panel of the compartment that has been blasted.

Both ".INR" files have the same format except that the interpretation of the records is different. Here is an example file, followed by an explanation of the record fields.

```
C INBLST RESULT FILE.                              DATE: Wed Apr 10 14:58:50 1996
C
C SHIP MODEL IS:
H                cube.INT
C RESULT FILE IS:
H                cube.INR
H  CUBE MODEL
C    27  TARGET BLOCKS IN DATA FILE
C      Compartment No.              Attack
PA     of Detonation:    20         Direction: STA
C
C      Area of Entry                Warhead
PB     Hole (m**2):  0.20           Weight (newtons):       400.00
C
PC     Aft to Fore:  0.50      Sta to Port:  0.30       Bottom to top:
0.20
C
C -----------------------------------------------------------------
C            DEST.     TIME   PRESS  VOLUME  PANEL    EVENT  SOURCE
C   PANEL COMPT.      (s)   (kpa)  (m**3) FAILURE    PROB  COMPT.  DIREC.
C   ----  ----   ------------  -----  -----  ------  ------- -----
RD     7     20  0.0000       0.      27.    1.00     1.00       0 POR
RD     2     19  0.0298     340.      27.    1.00     1.00      20 AFT
RD     8     23  0.0298     334.      27.    1.00     1.00      20 POR
RE    33      0  0.03       200.                      1.00       4  STA
```

The "PA", "PB", and "PC" records appearing in the header are output by *FBINBLAST* and read back by the display program *INGRAPH* for their display in the output window. They are used to pass the attack description exactly as read from the ".ATI" file.

Besides the header information, the *FBINBLAST* primary-damage file contains normally two kinds of records: "RE" and "RD". "RE" records are associated to external panels, "RD" records to internal panels. The first time a panel has a sufficiently high probability of rupture, a "RD" or "RE" record is output in the normal ".INR" file to reflect the current state. *INBLST* adds the newly invaded compartment to the list of invaded compartment, if applicable, and considers the possibility of subsequent ruptures.

As for "RD" records, the first field is the Integer ID of the panel that just ruptured, consistently with the information in the ".INT" file. The *DEST. COMPT.* field is the Integer ID of the compartment to which the blast is propagating and the *SOURCE COMPT.* field the ID of the source compartment. Both are identified consistently with the contents of the ".TBN"/".TBM" file; the two compartments are adjacent to the panel named in the *PANEL* field. *TIME* is the time in seconds where the panel had most chances to rupture.

VOLUME is the volume in cubic meters of the newly-added compartment. The meaning of the other fields depends on if the "INR" file is the normal one or is *Dispfile.INR*.

A. For the normal ".INR" file

PRESS is the acting gas overpressure in kilopascals at the moment of rupture. In this context, *pressure* means weighted pressure on the ruptured panel; the weights take into account the differential effects of the distances between the edges and the corresponding panel cell. *PANEL FAILURE* is the probability that the panel fails given the conditions that prevailed at the moment of rupture. *EVENT PROB.* is the cumulative probability that the new compartment (destination) be invaded by the blast.

B. For *Dispfile.INR*

PRESS is the maximum, throughout the simulation duration, of the mean gas over-pressure (KPa) in the invaded compartment; here, the mean is taken over the compartment's computational cells. *EVENT PROB.* is the maximal cumulative probability that the new compartment be invaded by the blast. *PANEL FAILURE* is the failure probability of the compartment's panel corresponding to the maximal cumulative probability of invasion (i.e. the last panel whose failure is possible).

In fact, for each possible destination compartment, *Dispfile.INR* contains a single record. It corresponds to the record of the ordinary ".INR" that has the same destination compartment and whose cumulative probability of invasion is maximal; if there are several candidates, the one corresponding to the latest time is chosen. Only the *PRESS* field of the *Dispfile.INR* record differs from the one of the latter.

In both cases, *EVENT PROB* = $PE(i,t)$ = cumulative probability of compartment i having been invaded by blast at time t ; it is defined recursively as follows:

$PE(i,t) = 0$ until the rupture probability threshold has been reached for at least one of the compartment's panel

At each new potential rupture, say across panel p, limiting compartments j and i, then $PE(i,t)$ becomes:

$$PE(i,t\text{-}dt) + PE(j,t\text{-}dt) \times Q(p,t)\,(1 - PE(i,t\text{-}dt))$$

where $Q(p,t)$ = current rupture probability of p, and where dt is the simulation step. This amounts to consider the current probability of compartment invasion at time t is the probability of the union of the two following events:

A = {i has already been invaded}
B = {i is presently being invaded across a new panel}

Note that the probability of *B* is nothing but the probability that the neighbor *j* has been invaded times the probability that the panel between *j* and *i* breaks, as indicated by the formula. A careful reader would have noticed that the calculation assumes that *A* and *B* are independent, which is false strictly speaking, but one wonders how to treat the problem without this assumption.

As for "RE" records, all the fields are the same as those of "RD" records, except for the addition of the *DIRECT* field and the suppression of the VOLUME and *PANEL FAILURE* fields, since *VOLUME* of the exterior of the ship does not make sense and since panel failure is equivalent to invasion of the exterior for an external panel. Note that the destination compartment is always *0*, meaning the exterior of the ship. The *DIRECT* field is a 3-character chain giving the external panel orientation with the following conventions:

POR - Port
STA - Starboard
FOR - Forward
AFT - Aft
TOP - Top
BOT - Bottom

To decide about panel rupture, *FBINBLAST* passes the ratio, *R* = (observed (corrected) pressure divided by the rupture-pressure) to a log-normal probability distribution, *gfunc2*, that estimates the rupture probability. To decide if the blast invades the compartment or not, which indirectly determines if possible ruptures of new panels in this compartment will be studied or not, the simulation compares the output value of *gfunc2* with a certain threshold (specified in the attack file) and proceeds to the rupture if the value of *gfunc2(R)* is higher, provided the response time has elapsed. When $R = 1$, the actual pressure has reached the theoretical rupture pressure; *gfunc2* then outputs *0.5* and increases or decreases very fast right or left, respectively, of this value. Such a log-normal algorithm is also used when estimating the component damage (program *DAMAGE*, routine *gfunc*). The graph of *gfunc/gfunc2* corresponds to a S-shaped increasing curve having value *0* at left of *0* and tending to *1* at *infinity*. The *COV* parameter of these routines controls the abruptness of the slope. *COV* may be interpreted as the squared root of $ERRP^2+EERR^2$ where *ERRP* and *ERRR* are the estimated relative errors (standard deviations or half *68%* confidence interval) on the rupture pressure and actual pressure, respectively. The two relative errors are assumed to be independent, normally distributed, with mean *0*. It is necessary to recompile *FBINBLAST* or *DAMAGE* , respectively, if *COV* is to be changed. See Chapter "DAMAGE component-Damage Program" of Ref. 8, for more details about the log-normal damage algorithm.

Obviously, the (cumulative or not) panel failure probabilities will always be higher than the value of the second field of the attack file "AL" record (safety factor). Experience shows that they are usually much higher, due to the abruptness of the log-normal damage function used.

3.2 INGRAPH Graphical Output

INGRAPH offers the user to visualize, on a deck-by-deck basis, the following information (see Fig. 6):

A: Maximal pressure
B: Prob event

MAXIMAL PRESSURE is the *PRESS* field of the "INR" file. Since the ".INR" file displayed should be *Dispfile.INR*, this corresponds to the maximum (over time) of the mean (over space) overpressure in the compartment.

Despite the name *Prob Event* that appears in the *INGRAPH* window, the color of each compartment does not represent a probability at all. It is the multiplication of the *PRESS* and *EVENT PROB* columns of *Dispfile.INR* , *EVENT PROB* being the cumulative probability of compartment invasion. *Prob Event*, to be distinguished from *EVENT PROB* may then be interpreted as the mathematical expectancy of the maximal overpressure in the invaded compartment (*DEST COMPT*).

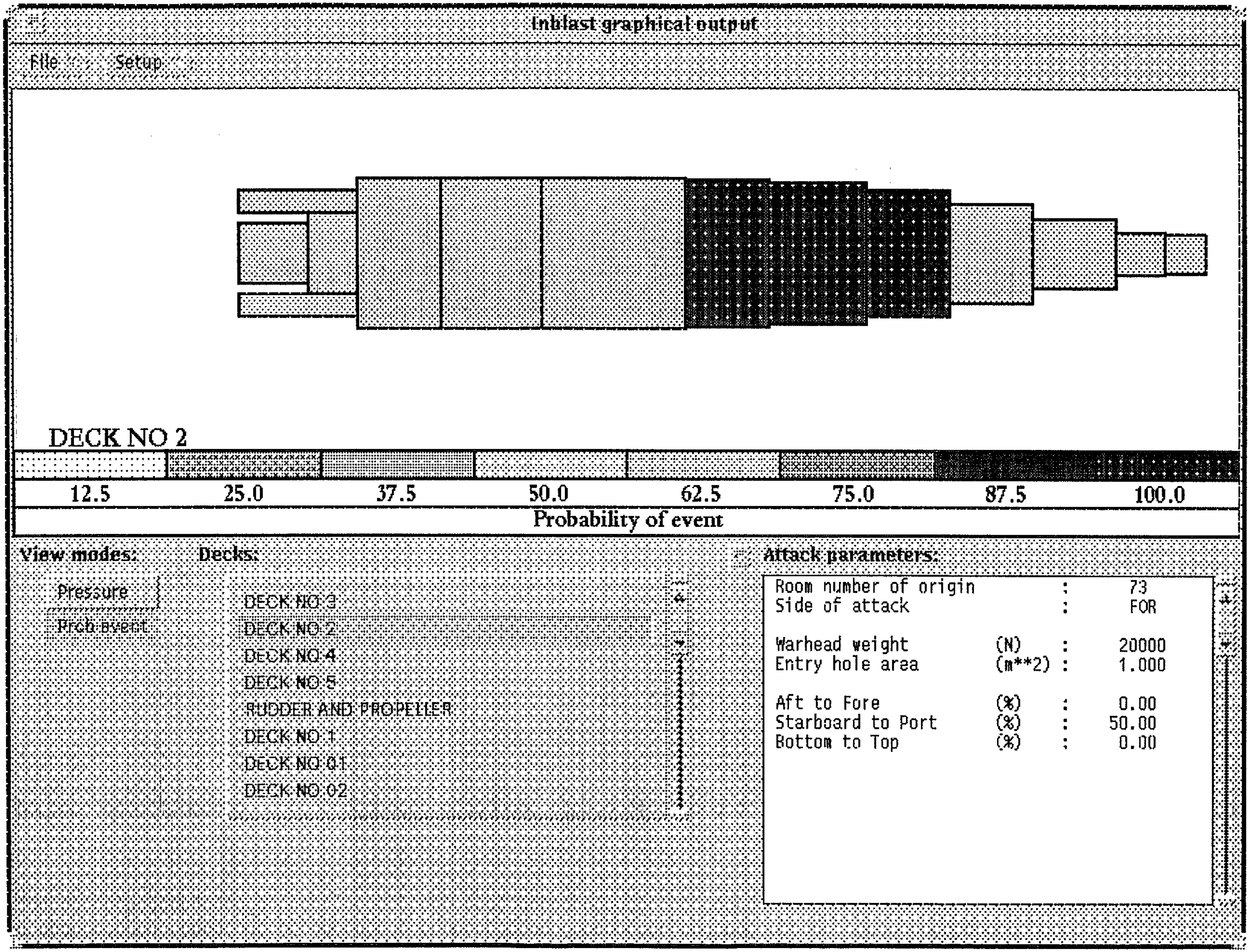

FIGURE 6 - INGRAPH Window

3.3 Component Vulnerability Results

If a component-damage analysis has been requested, the output will appear in a ".COMPDAM" file such as the following one:

```
C    COMPONENT DESCRIPTION FILE:  cube.fta
C
C    COMPONENT DAMAGE FILE:  cube.COMPDAM
C    DATE: Wed Apr 10 15:52:16 1996
C-----------------------------------------------------------------------
C
C
C  C               RECORD INDICATOR FOR A LINE OF COMMENTS
C  CD              RECORD INDICATOR FOR COMPONENT DAMAGE
C  CMPT ID         Compartment number in which considered component is
found
C  PANEL ID        ID of panel nearest from component in EXBLAST
C                  If component is NOT to be considered by EXBLAST, entry
C                  must be 0 (ZERO) else PANEL ID, GAP and SHOCK RESIST
C                  fields must be filled.
C  COMP ID         Component number to distinguish between components in
C                  same compartment
C
C  EXBLAST         Probability of component damage by exblast
C  INBLAST         Probability of component damage by inblast
C  FRAGMT          Probability of component damage by fragmt
C  FIRE            Probability of component damage by fire
C
C   IDENTIFICATION  |  P R O B A B I L I T Y   O F   D A M A G E
C  CMPT  PANEL COMP |    EXBLAST     INBLAST    FRAGMT      FIRE
C   ID     ID    ID |
C-----------------------------------------------------------------------
CD   20     0    -4     0.00000     0.99990    0.00000    0.00000
CD   21     0    -3     0.00000     0.00000    0.00000    0.00000
CD   19     0    -2     0.00000     0.99923    0.00000    0.00000
CD   19     0    -1     0.00000     0.99923    0.00000    0.00000
```

All the "C" records but those giving the names of the component/system description file and the name of the current ".COMPDAM" file consist in explanations concerning the different fields of the "CD" records. The "CD" records contain the component damage information *per se*.

There is one "CD" record for each component, damaged or not. The record contains the component output data in a format that should be transparent from the explanations appearing in commented form. Note that damage from other *GVAM* modules (External Blast, Fire and Fragmentation) can be combined with the internal blast damage, if desired (this is explained in Ref. 8, DAMAGE Chapter). The internal blast damage, that is calculated from the component descriptions of the current ".fta" file and from the primary damage output by *FBINBLAST* corresponds to the *INBLAST* column.

The *CMPT ID* field is the ID of the compartment containing the component, consistently with the compartment numbering scheme of the appropriate ".TBM" block file having been used to model the ship. *PANEL ID* is the ID of the panel that may have caused component damage due to external blast (nothing to do with FBINBLAST). The *COMP ID* field is the unique ID given to the component by the user while using the *Component* dialog of program *fta* ; it may be negative or positive, and gaps in the numbering scheme are allowed. The last four fields are the probabilities of component kill due to the external blast, internal blast, fragmentation and fire threats, respectively.

3.4 System Vulnerability Results

If a system-damage analysis has been requested, the output will appear in a '.fta-res' file such as the following one:

```
FTA RESULTS  file: cube.fta-res
    creation date: 10 Apr 1996

  FTA               file: cube.fta
                    date: 10 Apr 1996

  COMP-DAMAGE     file: cube.COMPDAM
                    date: 10 Apr 1996

  Damage from INBLAST
----------------------------------------------------------------
 PMA: 0 PMA
    SUB-PMA: Albert,
             weight: 0.25      survivability: 0.00
    SUB-PMA: Bert,
             weight: 0.25      survivability: 0.00
    SUB-PMA: Diane,
             weight: 0.25      survivability: 0.00
    SUB-PMA: Carl,
             weight: 0.25      survivability: 1.00
PMA survivability: 0.25
    rating:         M4     (Not ready)
----------------------------------------------------------------

System survivability: 0.25
Combat rating:          C4      (Not ready)
```

The file contains the system, subsystem and total (whole ship) vulnerability results. Prior to calculating system-vulnerability from the component-damage and the system descriptions of the ".fta" file, the component-damage from all the main threats (columns of the ".COMPDAM" file) that have been specified by the user are combined independently. It is the latter combination that is passed to program *SURV* to calculate the system-damage. See Ref. 8, to learn how to combine several damage mechanisms.

The header of the file contains the complete names of the files having been used to calculate the system vulnerability; first the name of the results file itself (.fta-res), followed by the name of the component/system-description file (.fta), the component-damage file (.COMPDAM) and the pre-damage file (.predam) if one has been used. See Ref. 8 for the meaning of pre-damage. The header ends with a line beginning with "Damage from", that gives the names of the up-to 4 main modules whose component-damage output has been combined to evaluate the joint component-damage. The possible names are: *FRAGMENT, EXBLAST, INBLAST, FIRE, PREDAMAGE*; the last name will appear only if a pre-damage file has been used and the others are main-module names.

Next comes the PMA/Sub-PMA survivability results. These are grouped by PMA with each PMA group beginning with a line starting with "PMA:" that gives the PMA number and name. PMA names are usually one of the following but the user may have defined any other while using the fta program:

AAW	*Anti Air warfare*
ASW	*Anti Submarine Warfare*
ASUW	*Anti Surface Warfare*
MOB	*Mobility*
CCC	*Communications Command and Control*
EW	*Electronic Warfare*
MW	*Mine Warfare*

The PMA report then lists all the Sub-PMAs (subsystems) constituting the PMA. To each Sub-PMA corresponds a line starting with "SUB_PMA:" followed by the name of the Sub-PMA and, on a new line, the weight (relative importance) of the Sub-PMA followed by its survivability. Note that the 'survivability' has to be considered as a survival probability or a Remaining Operational Capability (ROC), depending on the weighting scheme, as explained in Ref. 8.

Each PMA report ends with two lines of which the first gives the PMA (chain of Sub-PMAs) survivability, followed by a line listing a PMA rating between *M1* and *M4*, plus an explanation of the rating, between parentheses. The conventions for the ratings are as follows:

RATING	SURVIVABILITY	
M1	*90-100%*	*(fully ready)*
M2	*65-90%*	*(substantially ready)*
M3	*40-65%*	*(marginally ready)*
M4	*0-40% (not ready)*	

The total ship survivability is output at the end of the file on a line beginning with "System survivability" followed by a number. This line is followed by a last one entitled "Combat rating" which is the Combat Readiness Rating (CCR), between *C1* and *C4*. The possible values are:

RATING	STATE
C1	*Fully Ready*
C2	*Substantially Ready*
C3	*Marginally Ready*
C4	*Not Ready*

The formulas used to calculate each PMA survivability from the its sub-PMA survivabilities, and the rules for calculating the CCR from the PMA survivabilities consist in applying weighted mean algorithms abiding to NATO standards. The details are in Ref. 8, Chapter *fta*.

4. REMARKS

4.1 Changes from Previous Versions of FBINBLAST

As opposed to the previous versions of *FBINBLAST,* no more fudge factors ("AM" records) are required in the ".ATI" files

The commented panel lengths and widths in the ".INT"s, that were used in the previous versions are not needed anymore.

The holes corresponding to ruptured panels correspond to the whole panel span; on this point the former *INBLAST* was closer to reality since the hole sizes could have been specified in the ".INT" file. The finite-difference scheme imposes this constraint.

4.2 Execution Time

Execution time is about 30 min for the typical 0.06 sec. attack of 400N of TNT in compartment 20 of model cube.TBM (27 compartments).

4.3 Killing the Process

The program can be stopped, if necessary, with the command:

Kill process_number

(*CTRL C/D*) is not sufficient. The best policy is to run *FBINBLAST* in background mode and to use the returned process number for the kill. Alternatively use the *Unix* command *ps -aux* to find the process number.

The above is sometimes not enough to kill the process that still runs in background mode, often producing the system message:

program_name: Not enough memory,

while trying to start another program. Then use:

kill -9 process_number

4.4 Bug in F. Beaumont's ".INT" Files

There is a problem with the panel files that the contractor used for the trials. The problem occurs when modifying the files, even very slightly. The source of the problem may be the change of platform (*IBM Risc -> Sun Sparc*), possibly because of different conventions about carriage returns and line feeds to mark the end of lines. This problem does not occur when the files are generated using *gvmfm1* on *Sparc*.

4.5 Interpretation of the Maximal Overpressure.

The simulation end-time, especially if it is very short, may have a tremendous influence on the maximum (over time) of the mean overpressure in a compartment as output in *Dispfile.INR*. The reason is that the simulation could end much before the pressure had any chance to fill a newly invaded compartment; the output pressure could be lower than the maximum theoretically reachable.

APPENDIX B

TBM FILE FORMAT

The ".TBM" files describe the ship as a series of 3-D rectangular blocks. The file records also include panel thickness, type of material and stiffener setup information. The ship display program *ingraph* bases the deck-by-deck display on the ".TBM" file contents. Here is an example:

```
C      F R A G M E N T A T I O N    O N    A    T E S T    M O D E L
C      BASED ON DDH280
C-----------------------------------------------------------------------
H   CUBE MODEL
C      BASED ON DDH280
C
C  1.    COMMENT LINES MAY BE PLACED ANYWHERE IN THE FILE
C        BUT MUST START WITH A 'C' IN COLUMN 1
C  2.    INPUT VALUES MAY BE ENTERED IN A FREE FORMAT
C  3.    TARGET DESCRIPTION  MUST START WITH A 'UB' IN COLUMN ONE.
C  4.    COMMENTS (e.g. HEADINGS) TO BE PRINTED IN THE OUTPUT FILE
C        MUST START WITH A 'H' IN COLUMN 1
C
C
C         T  A  R  G  E  T     D  E  S  C  R  I  P  T  I  O  N
C-------------------------------------------------------------------------
C              WALL COORDINATES          WALL THICKNESS    MAT
PANELSTIFFENERS
C BLOC XMIN XMAX YMIN YMAX ZMIN ZMAX  X+ X- Y+ Y- Z+ Z-     SPAN HGT
THICK
C       cm   cm   cm   cm   cm   cm  mm mm mm mm mm mm   -   cm   mm   mm
C
UA DECK NO 3
C
UB 19    0  300    0  300  600  900  3  6  3  6  6  0   2   40   60   3
UB 20  300  600    0  300  600  900  3  0  3  6  6  0   2   40   60   3
.. etc
UB 27  600  900  600  900  600  900  6  0  6  0  6  0   2   40   60   3
C
UA DECK NO 04
C
UB 10   0   300    0  300  300  600  3  6  3  6  3  0   2   40   60   3
.. etc
UB 18  600  900  600  900  300  600  6  0  6  0  3  0   2   40   60   3

.. etc
```

Besides the comments and header records, two types of records, "UA" and "UB" are present.

"UA" records separate the compartment records (UB) into decks. Besides the record ID, the record has a single field, read as an 80-character chain, usually containing the deck number and a short description of it. The text is optional, but if it contains a *0* (ZERO) the deck will be considered as part of the superstructure; if no zero appears, the deck will be considered as part of the hull. The "UA" records will be rewritten without change into the ".TBN" files. This information will be utilized later, when the user chooses option 2 of *gvmfm1*, when it is time to define interactively the frequencies and rupture pressures of internal panels; this is how the program distinguishes between superstructure and hull panels. This means that it would be a good policy for the user to label the superstructure decks with numbers beginning with *0*, as illustrated by the second "UA" record above, and that *0*'s should not be used when labeling the hull decks, as in the first "UA" record above. The decks do not need to be ordered.

The "UB" records define the 3-D block characteristics. The different fields are as follows:

Compartment Number; no special ordering is necessary but the numbers must be unique.

Xmin, Xmax, Ymin, Ymax, Zmin and Zmax are Integers containing the respective compartment extensions (minimum and maximum) in the three canonical directions; units are centimeters and coordinates are in the ship coordinate system (see below). Note that min-type quantities should be smaller than the max-type quantities.

X+, X-, Y+, Y-, Z+ and Z- are Integers containing the respective panel thickness fields for the six panels in the compartment. The first letter in the field name refers to the direction of the normal to the panel (e.g. X corresponds to a panel in the YZ plane); field X+ is for the panel located at the XMAX coordinate, X- at XMIN, Y+ at YMAX, Y- at YMIN, Z+ at ZMAX and Z- at ZMIN. Units are millimeters. The coordinates conventions are as above.

Panel Material field: an Integer chosen between 1, for aluminum, and 2 (for steel).

The last three fields are the description of the panel stiffener setup in the compartment; all three are read as Floats. This information is not used by *FBINBLAST* (only by *EXBLAST*) but it is described below for completion. The fields are:

Stiffener-span, the distance (cm) between panel stiffeners.
Stiffener-height in mm (*HEIGHT* in Fig. 7).
Stiffener-thickness in mm (*TW* in the figure below)
The valid range is between *0* and *100* mm.

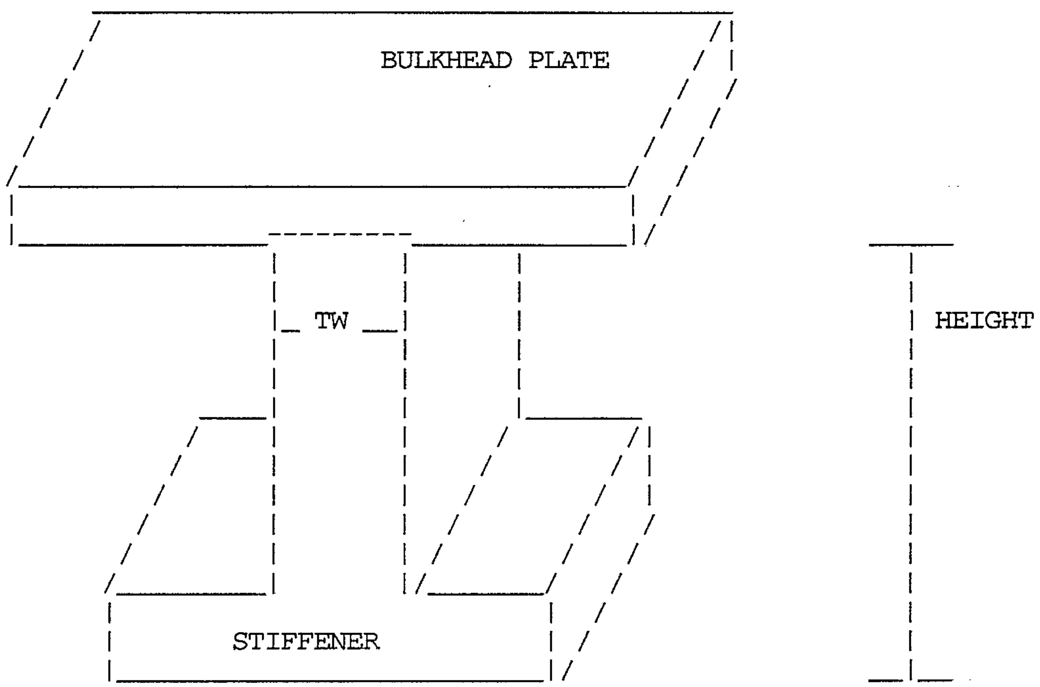

FIGURE 7 - Stiffener characteristics

Note that panel thicknesses from two adjacent internal walls (block borders) will be added to form what is called 'panel thicknesses' in ".INT" and ".EXT" panel files. This provides a way to play tricks like attaching a different thickness to the internal and external panels of a compartment. A thickness of *0* usually signifies that the panel thickness has been taken into account in the description(s) of the adjoining compartment(s); a compartment can also be defined with missing panels in the case of modelling a funnel or stack.

Coordinate System

The ship-based coordinate system is defined as follows: the positive Ox-axis runs along the centerline of the ship and originates at the rear of it; the Oz or vertical axis is positive up (Oz = 0 is the location of the waterline); the Oy-axis is positive in the port direction and its origin is at the center line of the ship. The aforementioned coordinates define a right handed coordinate system. Underwater decks are identified as those having a negative OZ coordinate (TBM/TBN files). The hull panels are those whose number (in UA record) begins with "0 (zero)". The decks should normally be named as follows (in UA records): hull decks are numbered consecutively 1, 2, 3 ... where 1 corresponds to the main deck and a higher number means a deeper deck; superstructure decks should be numbered consecutively 01, 02, 03, ... where the height of the deck increases with the index. The orientation conventions also apply to relative coordinates of components possibly included in compartments; the local origin is then the back starboard 0-height point. See Fig. 8 for an illustration of the conventions.

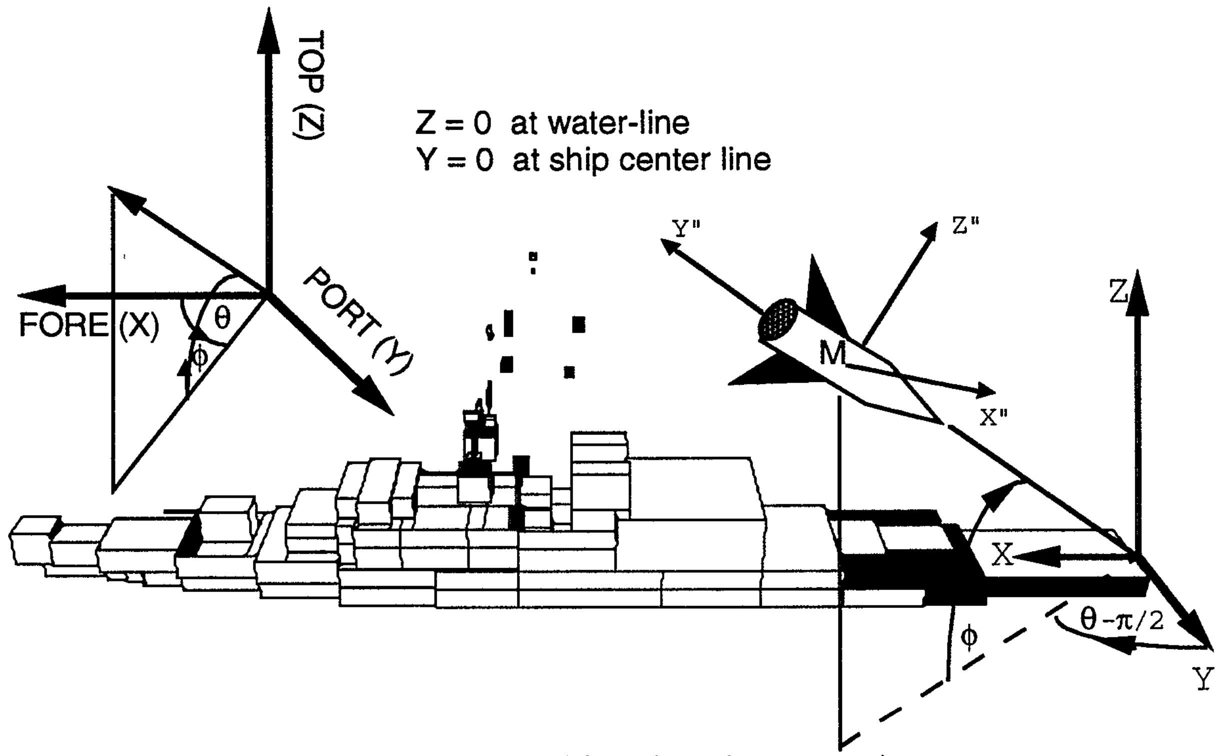

FIGURE 8 - FBINBLAST orientation conventions

APPENDIX C

PREPARING PANEL FILES FROM BLOCK FILES

The *gvam.CUR* scheme must be used in order to run *gvmfm1*. Generating panel files is done in two steps:

STEP 1: Change Directory to the one containing gvam.CUR and the I/O files (it must be the same). gvam.CUR must contain at least 'EB', 'EE' and 'EN' records, as illustrated below.

EB f1	*! "ship" internal panels (.INT):*
EE f2	*! "ship" three dimens. block model (.TBM):*
EN f3	*! INBLAST block model (.TBN):*
EC f4	*! "ship" external panels (.EXT):*

Call *gvmfm1* from a terminal window. Generate a ".TBN" file , here *f3.TBN*, from the *f2.TBM* file (interactive option 1). The output file name is requested interactively; normally, it must correspond to the contents of *gvam.CUR* (i.e.*f3* in the example). Example session:

```
albatross% gvmfm1

REMARKS: the former .TBM files must be converted into TBN format, with
variable stiffener span, before conversion into panel files. Please
CHECK that the new file is OK and add its name in file gvam.CUR if you
want to run GVMFM1 to produce panel files from it. The .INT file must
exist prior to panel waterbacking.

  MAKE A CHOICE BETWEEN:

  1: CONVERT .TBM files -> .TBN files
  2: CONVERT .TBN files -> .INT & .EXT files
  3: WATER-BACK UW panels and/or tanks
  4: CONVERT .TBN files -> RIPTAB-type .EXT files
  0: EXIT
1
  GIVE A FILE NAME FOR .TBN FILE OR <RETURN> IF IT IS: no_name_yet
  DO NOT ADD THE .TBN EXTENSION PLEASE
f2
The block definition records UB will be split in two; the second line
will contain the stiffener spans corresponding to each panel and a
common stiff. height & width. Edit the resulting file if the common
default value taken from the .TBM is not OK.

 PLEASE WAIT: creating file:CompMinus1.TBN
 STOP: End of GVMFM1
```

STEP 2: Call *gvmfml* once again to generate the '.INT' and '.EXT' files from f3.TBN file; this corresponds to interactive option 2 of *gvmfml;* the output files are named automatically from the '.TBN' file. The '.EXT' file is an external panel file not used by *FBINBLAST*, but *gvmfml* always generates one with any '.INT' file. Example:

```
albatross% gvmfm1

REMARKS: the former .TBM files must be converted into TBN format, with
variable stiffener span, before conversion into panel files. Please
CHECK that the new file is OK and add its name in file gvam.CUR if you
want to run GVMFM1 to produce panel files from it. The .INT file must
exist prior to panel waterbacking.

  MAKE A CHOICE BETWEEN:

  1: CONVERT .TBM files -> .TBN files
  2: CONVERT .TBN files -> .INT & .EXT files
  3: WATER-BACK UW panels and/or tanks
  4: CONVERT .TBN files -> RIPTAB-type .EXT files
  0: EXIT
2
 TYPE OF INPUT FOR PANEL FAIL.PRESSURES AND  EIGENFREQUENCIES

 1: INTERACTIVE, BASED ON PANEL ORIENTATION AND DECK
 0: CALCULATED FROM INFO IN .TBN FILE

 YOUR CHOICE ?
0
 The following info is needed for EXBLST only
 Enter INTEGER values only
 Span between frames (cm):100
 Span between stringers (cm):100
 Minimum allowable panel Len. or Width dimension (cm):50
 Span between frames (cm):   100
 Span between stringers (cm):   100
 Minimum allowable panel Len. or Width dimension (cm):     50
 Are the dimensions correct (Y/N)?y
STOP: End of GVMFM1
```

The requested 'Span between frames and stringers' as well as the 'Minimum allowable panel Len. or Width' must be less than the smallest box dimension. The other options offered by *gvmfml* are not to be used in the *FBINBLAST* context. More information is available in Ref. 8, Chapter "GVMFM1 FILE CONVERSION PROGRAM", if needed.

APPENDIX D

INT FILE FORMAT

Here is an example of an ".INT" file, as used by *INBLST* and *FBINBLAST*. The file is normally created by *gvmfml* from a ".TBN" file plus some user input information, but it may be edited subsequently.

```
C        Tue Jul 13 09:05:28 1993
C Inblast formatted input, generated from :
C         ddh280.TBN
C
C  Span between frames (cm):  400
C  Span between stringers (cm):  200
C  Minimum allowable panel Len. or Width dimension (cm):   20
C
H  FRAGMENTATION ON A DDH-280
C
C  -----------------------------------------------------------
C  CONTAINS ROOM AND WALL DESCRIPTIONS
C  FILE BASED ON TARGET DESCRIPTION (T FILE)
C
C  ]-------]-------]-------------------------------]
C  ] COMPT.] VOLUME]  COMPARTMENT DESCRIPTION       ]
C  ]  No.  ] (m**3)]                                ]
C  ]       ]       ]                                ]
C  ]-------]-------]-------------------------------]
RA       1    239.80
RA       2    493.44
.
.
C          P A N E L   A R R A Y
C
C  ]-------]-------]-------]-------]-----------]--------]----------]
C  ] PANEL ] COMPT.] COMPT.] PANEL ]    PANEL  ]NATURAL ] LOCATION ]
C  ]  No.  ] No.1  ]  No.2 ] VENT. ]   FAILURE ]  FREQ. ]          ]
C  ]       ]       ]OUTSIDE] AREA  ]    PRESS. ]        ]          ]
C  ]       ]       ]   =0  ] (m**2)]    (kpa)  ]  (Hz)  ]          ]
C  ]-------]-------]-------]-------]-----------]--------]----------]
RB         1      1      2    0.00        90.0      26.0     AFT
RB         2      1      0    0.00       130.0      40.0     AFT
.
etc
```

The information about frame and web spacing and minimal panel dimensions at the beginning of the file is written by *gvmfml* and corresponds to the user's input described in Appendix C.

The "RA" records are used for compartment description. *COMPT. No* is the compartment number and must be consistent with the corresponding block ID appearing in the ".TBN" file. *VOLUME* is the value of the compartment volume in cubic meters.

The "RB" records contain the panel-related information. *Panel No.* is the ID of the panel. *COMPT No. 1* and *COMPT No. 2* are the IDs for the two compartments adjacent to the panel. A panel is external if and only if *COMPT No. 2* is *0*. The conventions when one talks about the compartment neighbors of a panel, WE1 and WE2, are the following ones: the direction pointing from WE1 toward WE2 corresponds to the panel orientation. More precisely, the following table identifies on which side of the panel is each of the two neighbors:

Panel Orientation	WE1	WE2
FOR	FOR	AFT
AFT	AFT	FOR
TOP	TOP	BOT
BOT	BOT	TOP
POR	POR	STA
STA	STA	POR

See below for the meaning of the 3-character chains.

The next three fields contain the panel vent area in square meters, the panel-failure-pressure in kilopascals and the panel natural frequency in hertz. The vent area will be *0* unless the file has been manually edited by the user; this information is irrelevant for *FBINBLAST* anyway, since the whole panel is the vent area in case of rupture. The last two of these fields, the panel's eigenfrequency (in hertz), and rupture-pressure (in kilopascals) have normally been calculated or user-input in *gvmfm1*.

LOCATION is a three-character chain representing the panel orientation with respect to *COMPT 1*; the conventions below apply:

```
POR - Port
STA - Starboard
FOR - Forward
AFT - Aft
TOP - Top
BOT - Bottom
```

As explained in file "gvam/GVMFM1/docs/Readme93", the fields possibly following the '!' character at the end of any "RD" record, are the geometrical panel dimensions (as opposed to the stiffener-limited dimensions used for panel-response). They are normally written by program *gvmfml* when generating the panel rupture pressures and eigenfrequencies. This is not used by *FBINBLAST* anymore and is considered as comments.

Note that *gvmfml* replaces records identifiers of type "RB" with "C" record identifiers when the panel height or width are small enough (this has nothing to do with panel thickness). These commented panels are considered non-existent since the corresponding records are not read-back by *FBINBLAST*.

A compartment's wall can be made of several panels, normally at least as many as the number of compartments distinct from the current one that are adjacent to the wall.

Panel thicknesses from two compartments adjacent to a common panel are added by *gvmfml*.

FBINBLAST identifies dummy panels by the fact that their eigenfrequencies and failure pressures are *0* in the ".INT" file. Program *gvmfml* generates such a panel when the corresponding panel has a zero thickness following the information in the ".TBN" file. Dummy panels are broken immediately after the onset of their pressurization in *FBINBLAST*.

APPENDIX E

FBINBLAST LOGICAL TREE

```
FBINBLAST                                  [FBINBLAST_95]Inblst.for
    |
    |__GVAM_CUR                            [UTIL]GVAM_cur.for
    |       |
    |       |__GET_CUR_FILENAME            [UTIL]GVAM_cur.for
    |       |       |
    |       |       |__PARSE               [UTIL]Parse.for
    |       |
    |       |__ADD_EXTENSION               [UTIL]GVAM_cur.for
    |
    |__PARSE                               [UTIL]Parse.for
    |
    |__DECDVK                              [UTIL]Decdvk.for
    |
    |__DECDDA.FOR                          [UTIL]Decdda.for
    |
    |__FBREADM                             [FBINBLAST_95]FBReadm.for
    |       |
    |       |__PARSE
    |       |
    |       |__DECDDA
    |       |
    |       |__DECDRA                      [UTIL]Decdra.for
    |       |       |
    |       |       |__FNDDEC              [FBINBLAST_95]Fnddec.for
    |       |
    |       |__DECDRB                      [UTIL]Decdrb.for
    |       |       |
    |       |       |__FNDDEC
    |       |
    |       |__CHECK                       [FBINBLAST_95]Check.for
    |
    |__SIMUL                               [FBINBLAST_95]Simul.for
    |       |
    |       |__MISEO                       [FBINBLAST_95]Miseo.for
    |       |
    |       |__TINPUT                      [FBINBLAST_95]Tinput.for
    |       |
    |       |__SETTING                     [FBINBLAST_95]Setting.for
    |       |
    |       |__CALCOEF                     [FBINBLAST_95]Calcoef.for
    |       |
    |       |__LOCATE                      [FBINBLAST_95]Locate.for
    |       |       |
    |       |       |__PARSE
    |       |       |
    |       |       |__DECDAH              [UTIL]Decdah.for
```

```
|          |          |          |
|          |          |          |__FNDDEC
|          |          |
|          |          |__DECDAI                    [UTIL]Decdai.for
|          |          |          |
|          |          |          |__FNDDEC
|          |          |
|          |          |__DECDDA
|          |
|          |__ADDITION                     [FBINBLAST_95]Addition.for
|          |          |
|          |          |_TYPEINIT                   [FBINBLAST_95]Typeinit.for
|          |
|          |__BLASTINIT                    [FBINBLAST_95]Blastinit.for
|          |
|          |__CONDFRON                     [FBINBLAST_95]Condfron.for
|          |
|          |__FLUX                         [FBINBLAST_95]Flux.for
|          |
|          |__FLUXXO2                      [FBINBLAST_95]FluxxO2.for
|          |
|          |__FLUXXO3                      [FBINBLAST_95]FluxxO3.for
|          |
|          |__FLUX                         [FBINBLAST_95]Flux.for
|          |
|          |__FLUXYO2                      [FBINBLAST_95]FluxyO2.for
|          |
|          |__FLUXYO3                      [FBINBLAST_95]FluxyO3.for
|          |
|          |__FLUX                         [FBINBLAST_95]Flux.for
|          |
|          |__FLUXZO2                      [FBINBLAST_95]FluxzO2.for
|          |
|          |__FLUXZO3                      [FBINBLAST_95]FluxzO3.for
|          |
|          |__CALPRES                      [FBINBLAST_95]Calpres.for
|          |
|          |__RUPTURE                      [FBINBLAST_95]Rupture.for
|                     |
|                     |__ADDITION
|                                |
|                                |_TYPEINIT
|
```

```
|
|__DAMAGE                              [FBINBLAST_95]Damage.for
        |
        |__PARSE
        |
        |__DECDVR                      [UTIL]Decdvr.for
        |       |
        |       |__FNDDEC
        |
        |__DECDDA
        |
        |__DECDVB                      [UTIL]Decdvb.for
        |
        |__GFUNC2                      [FBINBLAST_95]Gfunc.for
        |
        |__EVAL                        [FBINBLAST_95]Eval.for
```

UNCLASSIFIED

INTERNAL DISTRIBUTION
DREV R-9612

1 - Deputy Director General

1 - Chief Scientist

6 - Document Library

1 - C. Fortier (author)

1 - Head, Weapons Effects Section

1 - Leader, Vulnerability/Lethality Group

1 - D. Bourget

EXTERNAL DISTRIBUTION
DREV R-9612

2 - DRDIM

1 - CRAD

1 - DMOR

1 - DSAM 8

1 - DSE 6-4-2
Attn: M. Belcher

1 - DRES Military Engineering Section
Attn: J. Slater

1 - François Beaumont (co-author)
3275 Lacombe
Montréal, H3T 1L6

1 - Prof. J.Y. Trépanier
École Polytechnique de Montréal
CP. 6079 succursale A
Montréal, H3C 3A7

1 - Navware Canada Inc.
105 Côte de la Montagne, Suite 701
Québec, G1K 4E4
Attn: P. Barbeau

AUSTRALIA

1 - Dr. Norbert Burman
Materials Research Laboratories
P.O.Box 50, Ascot Vale
Victoria, 3032
Australia

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

1. ORIGINATOR (name and address) DREV	2. SECURITY CLASSIFICATION (Including special warning terms if applicable) UNCLASSIFIED

3. TITLE (Its classification should be indicated by the appropriate abbreviation (S,C,R or U)
A CFD COMPUTER SIMULATION FOR MODELLING LARGE-SCALE BLAST PROPAGATION (FBINBLAST)

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)
Fortier, C. and Beaumont F. * Ecole Polytechnique de Montréal

5. DATE OF PUBLICATION (month and year) September 96	6a. NO. OF PAGES 53	6b. NO. OF REFERENCES 15

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. Give the inclusive dates when a specific reporting period is covered.)
DREV Report

8. SPONSORING ACTIVITY (name and address)
TNS 2BA14 (Vehicle Damage & Crew Casualties Modelling)

9a. PROJECT OR GRANT NO. (Please specify whether project or grant) PCN 21B14	9b. CONTRACT NO.
10a. ORIGINATOR'S DOCUMENT NUMBER R-9612	10b. OTHER DOCUMENT NOS. N/A

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

- [X] Unlimited distribution
- [ ] Contractors in appoved countries (specify)
- [ ] Canadian contractors (with need-to-know)
- [ ] Government (with need-to-know)
- [ ] Defence departments
- [ ] Other (please specify) :

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report describes a large-scale medium-resolution computer simulation of blast propagation inside a structure with many compartments, following the explosion of a conventional warhead inside one of the compartments. The simulation is an extension of the quasi-static blast-propagation simulation INBLAST, which is part of the General Vulnerability Assessment Model. In contrast with the 1-D or 0-D former INBLAST, the new simulation uses a 3-D third order CFD scheme to simulate the propagation of the explosion products from compartment to compartment and is able to model the shock wave in addition to the quasi-static pressure. The rupture pressures and the systems' vulnerability are computed in a way that is similar to what was done in the former version. Refinements in the modelling of panel response and rupture have been added in order to better utilize the more precise loading characterization of the new CFD algorithm.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each sould be indicated as with the title.)

blast
Inblast
Inblst
GVAM
shock wave
vulnerability
lethality
simulation

S01929

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

(DCD03E.IFD - 95.02.22)